

UNITED STATES AIR FORCE RESEARCH LABORATORY

Air Force ALP AEF Multi-Planning Development and Demonstration

Nicholas J. Stute
Christopher S. Allen
Cynthia K. Colby

TASC, Inc.
2555 University Blvd.
Fairborn, OH 45324

Christopher K. Curtis
Air Force Research Laboratory

March 2001

Final Report for the Period March 2000 to February 2001

20010802 044

Approved for public release; distribution is unlimited.

Human Effectiveness Directorate
Deployment and Sustainment Division
Logistics Readiness Branch
2698 G Street
Wright-Patterson AFB OH 45433-7604

NOTICES

When US Government drawings, specifications or other data are used for any purpose other than a definitely related Government procurement operation, the Government thereby incurs no responsibility nor any obligation whatsoever, and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications or other data, is not to be regarded by implication or otherwise, as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

Please do not request copies of this report from the Air Force Research Laboratory. Additional copies may be purchased from:

National Technical Information Service
5285 Port Royal Road
Springfield, VA 22161

Federal Government agencies registered with the Defense Technical Information Center should direct requests for copies of this report to:

Defense Technical Information Center
8725 John J. Kingman Rd., Ste 0944
Ft. Belvoir, VA 22060-6218


TECHNICAL REVIEW AND APPROVAL

AFRL-HE-WP-TR-2001-0076

This report has been reviewed by the Office of Public Affairs (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

FOR THE COMMANDER


ALBERT S. TORIGIAN, Lt Col, USAF
Deputy Chief
Deployment and Sustainment Division
Air Force Research Laboratory

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
<small>Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.</small>				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE March 2001	3. REPORT TYPE AND DATES COVERED Final - March 2000 - February 2001	
4. TITLE AND SUBTITLE Air Force ALP AEF Multi-Planning Development and Demonstration			5. FUNDING NUMBERS C: F33615-99-D-6001 DO: 10 PE: 63106F PR: 2745 TA: 00 WU: 31	
6. AUTHOR(S) Nicholas J. Stute, Christopher S. Allen, Cynthia K. Colby, Christopher K. Curtis				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) TASC, Inc. 2555 University Blvd. Fairborn, OH 45424			8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Air Force Research Laboratory, Human Effectiveness Directorate Deployment and Sustainment Division Air Force Materiel Command Logistics Readiness Branch Wright-Patterson AFB, OH 45433-7604			10. SPONSORING/MONITORING AGENCY REPORT NUMBER AFRL-HE-WP-TR-2001-0076	
11. SUPPLEMENTARY NOTES				
12a. DISTRIBUTION AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) Doctrinally, Joint Vision 2020 and Focused Logistics, as well as the Air Force's vision for Global Engagement and Agile Combat Support, chart a course for the future of logistics as a foundation to operations support. These visions place great demands on the logistician and logistics infrastructure to provide rapid, scaled, and agile logistics to the operator worldwide. To meet those demands, the Logistician requires enhanced information support for logistics command and control issues. Early evidence suggests that one of the major constraints in achieving Agile Combat Support and short-notice expeditionary operations is our Logistics Command and Control (Log C2) infrastructure. The AF requires an infrastructure that facilitates force support planning, execution, and monitoring across all operational phases of all missions. The purpose of this task is to develop a cluster society based on DARPA's Advanced Logistics Project (ALP) infrastructure. This is a continuation of the GSA contract titled ALP AEF Initiative (F41624-99-F-0003). The primary focus of this delivery order is to add additional functionality that was developed under the previous effort. This effort will focus on adding functionality for supporting multiple concurrent operations plans (OPLANS).				
14. SUBJECT TERMS Advanced Logistics Project (ALP) Air Expeditionary Force (AEF) Command and Control Agile Combat Support Logistics Defense Advanced Research Projects Agency (DARPA)			15. NUMBER OF PAGES 38	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT UL	

THIS PAGE LEFT INTENTIONALLY BLANK

PREFACE

This technical report contains the results of the Air Force ALP AEF Multi-Planning Development and Demonstration task. This work was executed on the Air Force Research Laboratories' Technology for Readiness and Sustainment (TRS) contract. This effort was Delivery Order number 10 and the contract number was F33615-99-D-6001. The work described in this report was performed during the period 29 March 2000 through 28 February 2001. The objective of this task was to enhance and expand upon the previous effort of applying the Defense Advanced Research Projects Agency's (DARPA) Advanced Logistics Project (ALP) architecture to model a subset of the logistics operations needed to support the deployment and sustainment of Air Expeditionary Force units (AEF). Advanced Logistics Project Integration and Engineering (ALPINE) is executing the development of the ALP architecture, a joint venture between GTE- BBN Technologies and Raytheon Systems.

The principal investigators for this effort included Mr. Chris Curtis, Capt. Adrian Crowley, and Capt. David Sanford from AFRL/HESR, Mr. Nick Stute, Mr. Chris Allen, and Ms. Cynthia Colby from Litton TASC Inc.

Table Of Contents

PREFACE	iii
LIST OF FIGURES	v
ACRONYMS	vi
Introduction to the ALP Program.....	1
Background	1
Key Features of the ALP Architecture	2
ALP Cluster Concepts	3
LogPlan	4
PlugIns.....	5
Expander Plugin	5
Allocator Plugin	5
Assessor Plugin	6
Logical Data Model (LDM) Plugin.....	7
Plan Service Provider (PSP) PlugIn	7
Cluster Relationships.....	7
Putting the Pieces Together.....	8
ALP's Decision-Making Philosophy	9
ALP Development Environment.....	9
Yearly Demonstration Scenarios.....	10
Future of ALP.....	11
AEF Community – Logistics Planning for AEF Deployments.....	12
Development Progression of the AEF Community of Clusters	12
Synopsis of this Year's Work.....	16
Descriptions of Organizations Modeled in 2000 AEF Community	16
High Level Task Flow in the 2000 AEF Community	17
Operations and Logistics Synergy.....	19
User Interfaces.....	20
Generic Functionality and Integration with other ALP Communities	26
Generalized Tools and the Generic Logistics Model (GLM).....	27
Description of Generic AEF Functionality	27
Contributions to the GLM	27
Final Year of ALP Development	28
Conclusions	29
References	31

List Of Figures

Figure 1: Cluster Components.....	4
Figure 2: Plugin Operation Flow.....	8
Figure 3: Year One AEF Cluster Community.....	13
Figure 4: Year Two AEF Cluster Community.....	14
Figure 5: Year Three AEF Cluster Community.....	15
Figure 6: Key Components in the AEF Community.....	17
Figure 7: Ops/Log Synergy.....	20
Figure 8: Force Package Mix Results.....	21
Figure 9: Create Force Package Mix.....	21
Figure 10: Squadron Aircraft Schedule.....	22
Figure 11: Aircraft Historical Data.....	23
Figure 12: Chalk View User Interface.....	24
Figure 13: AEF Society Completion User Interface.....	25
Figure 14: Asset Sourcing Results User Interface.....	26

ACRONYMS

AEF	Air Expeditionary Force
AFIT	Air Force Institute of Technology
AFRL	Air Force Research Laboratory
ALP	Advanced Logistics Project
ALPINE	Advanced Logistics Project Integration and Engineering
API	Application Programmers Interface
COUGAAR	Cognitive Agent Architecture
DARPA	Defense Advanced Research Projects Agency
DLA	Defense Logistics Agency
DOD	Department of Defense
GLM	Generic Logistics Model
HTML	Hypertext Markup Language
HTTP	Hypertext Transport Protocol
IDE	Integrated Development Environment
JDK	Java Development Kit
LDM	Logical Data Model
M-R VAT	Mission-Resource Value Assessment Tool
NSN	National Stock Number
PSP	Plan Service Provider
SEAD	Suppression of Enemy Air Defense
SOW	Statement of Work
TRANSCOM	Transportation Command
TRS	Technology for Readiness and Sustainment
USAF	United States Air Force
UTC	Unit Type Code
WRM	War Reserve Material
XML	Extensible Markup Language

INTRODUCTION TO THE ALP PROGRAM

Background

The Advanced Logistics Project (ALP) is a five-year, multi-phased advanced research project jointly sponsored by the Defense Advanced Research Projects Agency (DARPA) and the Defense Logistics Agency (DLA). This program is investigating, developing, and demonstrating technologies that will make a fundamental improvement in logistics planning and execution efficiencies. It is defining, developing, and demonstrating advanced technologies that enable forces and sustainment materiel to be deployed, tracked, sustained, refurbished, and redeployed more efficiently and effectively than ever before, during peacetime and contingency operations.

The current logistics operating environment uses isolated, independent, and sometimes incompatible systems, processes, and data. As a result, planning lacks realistic detailed data necessary to provide effective and timely logistics support at the lower levels of command; higher levels of command lack visibility into ongoing logistics operations at lower levels of command; and there is no common interoperable end-to-end systems view upon which decision-makers, at any level, can rely. Consequently, the very rapid replanning and redirection necessary to support crisis action responsiveness for multiple simultaneous missions is challenging to accomplish with the systems in place today.

This program addresses these and other shortcomings of existing logistics support systems and seeks full development of significantly improved capability. The effort has large potential cost savings for both the federal and private sectors through greatly improved management of manufacturing, storage, transportation and repair assets. Development of automated, multi-echelon, real-time collaborative technologies for the joint logistics communities is intended to provide logisticians and war fighters an unprecedented capability to plan, execute, monitor, rapidly replan and re-execute logistics support, even while assets are enroute to the theater of operation.

Key Features of the ALP Architecture

The ALP system is highly automated. Logistics decision-making logic, the capability to assign equipment and personnel, and the capability to schedule resources and transportation are programmed into the ALP system components. Although automated, users have an important role in the executing ALP system. Users provide the policies and rules that govern these processes, can approve decisions made by the system, intervene when necessary to resolve conflicts, and provide solutions for shortfalls and issues with time constraints. Users can also inject tasks into the system for actions that may not have been considered by the system designers, or to incorporate real world events into the processes.

The ALP system is a highly distributed system comprised of many *clusters*. A cluster is a portion of an ALP “society” representing the domain logic of a particular organization, such as Transportation Command (TRANSCOM), DLA, or perhaps a maintenance group or a base hospital. The scope of a cluster’s functionality is quite flexible. A cluster’s functionality can be specific to a small sphere of activities or processes, such as the assignment of housing at a base. Or, the functionality may be more encompassing, such as the entire process of deploying a combat mission. The collection of all ALP clusters is called the ALP *society*. A subset of clusters makes up a *community*. For example, the set of clusters representing the functionality at a fighter wing could be considered a community. Note that a given community could have a great deal of processing that is done independently from the ALP society as a whole, e.g., allocating and scheduling resources for training missions, periodic maintenance, food supplies, etc. The same community could also be involved in a society-wide scenario, such as the deployment of forces and equipment for a particular mission.

ALP provides the definition of standard communications protocols that sit on top of the network hardware infrastructure. Clusters provide different types and levels of information, but the ALP architecture ensures that all “clusters speak the same language” so they can exchange information or services.

Based on having a standard set of communication messages between clusters, groups of ALP clusters can be set up to operate with each other, sharing information seamlessly, and requiring

very little human intervention to facilitate the communication process. Information, which currently may take several phone call inquiries to different United States Air Force (USAF) installations, would, in most situations, flow automatically to the clusters requiring it. Automatic updates of data would be provided as real world situations change. The potential seamless and near real time dissemination of messages between clusters could provide the logistician with a more complete up-to-date picture of the state of an operation or for the planning of an operation than is currently available.

The ALP system provides the capability for continuous replanning and updating of the logistics plan. The status of real world events can change the availability of resources, or can change the priority of future events. With the ALP system, the availability or status of resources can be monitored and allocated/reallocated to improve timeliness, cost, effectiveness, or to minimize loss of life, etc. These changes can be handled by cluster functionality, resulting in changes to allocated resources, or can cause elements of the cluster's LogPlan to be changed or new elements added. The ALP infrastructure automatically propagates these changes to other clusters, where subsequent plan alterations may be initiated.

Moreover, ALP supports the combination of planning and execution requirements. As time passes, planned events become reality, and then become artifacts of the past. The results of those events can have an effect on future events. The policies and functionality to support this replanning can be incorporated into the logic of the clusters.

ALP Cluster Concepts

Figure 1 shows the basic components of a cluster. The ALP application programming interface (API) defines the methodologies for clusters to communicate with each other and the methodologies for each cluster to communicate information within itself. A cluster makes requests of other clusters via outgoing tasks and receives requests from other clusters via incoming tasks. Different clusters can be resident on the same machine or reside on different machines.

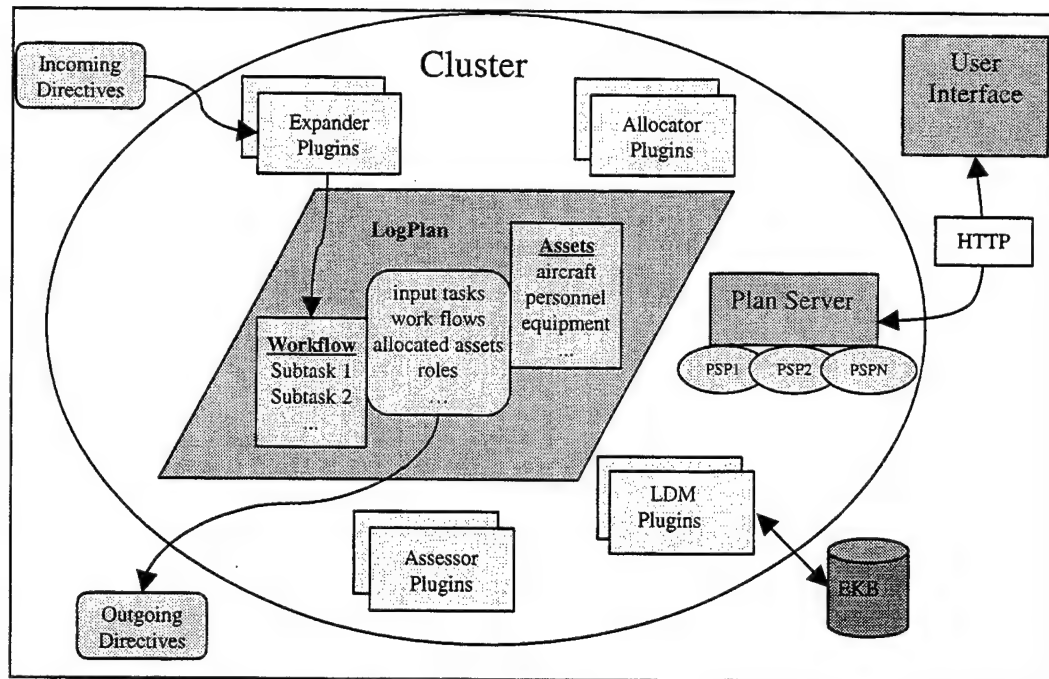


Figure 1: Cluster Components

A cluster consists of ALP LogPlan elements, a combination of various types of *plugins*, and, in order to fit into the society, portions of the ALP infrastructure. Plugins are modular, self-contained discrete units that provide the domain specific functionality of the cluster. It is intended that, as functionality is created, it can be “plugged into” the cluster, giving the cluster the ability to handle more tasks. As functionality is maintained (fixes, enhancements, etc.), plugins can be “pulled out” and replaced with updated ones, without interrupting the rest of the system’s processing.

LogPlan

Consider the section in Figure 1 labeled “LogPlan.” Each ALP cluster contains a LogPlan that reflects the processes and planning accomplished by that cluster. In actuality, this is the collection of all of the cluster’s data including, among other things, the cluster’s assets, input tasks, workflows, its relationships with other clusters, and the actual logistics plan elements. Note that each cluster in the ALP society maintains its own LogPlan. The LogPlan maintains the state of a cluster. The capability to persistently store LogPlans was developed by the ALPINE team during the 1999 ALP architecture development. Java’s serialization mechanism was leveraged for implementing this functionality.

All of the plugin types can communicate directly with the cluster's LogPlan. Depending on the situation, this link may be for read-only functionality. For example, Plan Server PlugIns (PSP) may be created to expose various views of the logistics information, such as asset usage, or schedules. On the other hand, when an expander plugin creates a workflow during a task expansion, it writes the workflow directly to the LogPlan. An allocator plugin finds new workflows and available assets in the LogPlan and submits asset allocations, schedules, and penalty values back to the LogPlan.

PlugIns

Figure 1 shows several types of plugins associated with a cluster: Expander, Allocator, Assessor, Logical Data Model (LDM) and PSP. The following sections provide a description for each of the plugin types.

Expander Plugin

An expander plugin, also called a task expander, performs the initial processing of each input task received by the cluster. This plugin expands input tasks into one or more subtasks that the cluster knows how to complete. Each input task's set of subtasks is referred to as a *workflow*.

For example, the input task "Generate AEF using OPLAN10A" might be expanded into a workflow containing the following subtasks:

Subtask 1 = "Determine requirements for Suppression of Enemy Air Defenses (SEAD) using OPLAN10A"

Subtask 2 = "Determine requirements for Air Interdiction using OPLAN10A"

Subtask 3 = "Determine requirements for Air Superiority using OPLAN10A"

Allocator Plugin

A cluster's assets can include both physical assets and other cluster assets. It is the responsibility of an allocator plugin to allocate the cluster's assets to complete the subtasks of each workflow. The allocator plugin may also choose to delegate the responsibility for completing a subtask to another cluster, which is the principal means of creating an output task. The plugin would choose

the target cluster for this kind of directive by means of inter-cluster *relationships* and cluster *capabilities* or *roles*. These concepts are discussed later in the section titled “Cluster Relationships.”

It is the allocator’s function to maintain the “best” allocation of the cluster’s assets. That is, as the plugin considers assets for new workflows, it may be necessary to reallocate existing asset allocations in order to improve the overall usage of its resources. For example, suppose an allocator plugin has designated a particular truck to perform a transportation request. Now suppose the cluster receives another transportation request. It may be more economical to deallocate the first truck and allocate a single larger one to handle both transportation requests, than to allocate a second truck dedicated solely to the new transportation request. This type of logic would have to be built into an allocator plugin used by that cluster.

In addition to allocating the cluster’s assets, it is the allocator’s responsibility to assign schedules of usage and to satisfy task constraints while considering *penalty values* for its allocations. Task constraints are rules that affect how performing one task affects the performance of other tasks. For example, it may be required that one task is completed before another task is started. Penalty values represent the requester’s view of the importance of a task. These features of the ALP architecture have the potential to provide an ALP-based planning system with high fidelity and flexibility by incorporating decisions based on various priorities between tasks.

Assessor Plugin

An assessor plugin is responsible for monitoring the execution of the plan. By considering real world events, including satisfactory completion of projected plan components, as well as verifying overall objective conformance, an assessor can watch for plan deviations. The plugin may incorporate various thresholds to ensure that successful plan execution is not jeopardized. Assessor plugins can generate exceptions to alert appropriate mechanisms that remedial actions may be required or may simply insert new tasks into the system to directly effect replanning processes.

Logical Data Model (LDM) Plugin

LDM plugins are responsible for mapping contemporary data into the ALP society. This is the means for providing an ALP wrapper for existing data sources. It is the LDM plugin's responsibility to maintain interfaces with its data sources and to act as a liaison between ALP processes and the processes that natively work with each of its data sources. For example, this could include updates to contemporary databases due to ALP processing or may include updates to ALP processing due to triggers set up in the databases.

Plan Service Provider (PSP) PlugIn

The primary goal of the PSP mechanism is to expose the contents of the LogPlan through a network protocol for purposes of creating user interfaces and to provide external systems with internal details of ALP planning processes. A hypertext transfer protocol (HTTP) server (typically referred to as a web server) resides on each cluster. The user creates PSPs, which are accessed through the web server running on the cluster. Any client needing information about the cluster can communicate with the PSP using standard HTTP protocol to query for specific LogPlan information. The PSP has most of the same privileges as a normal plugin; in particular it can query the contents of the LogPlan. The PSPs are written to produce Hypertext Markup Language (HTML), Extensible Markup Language (XML) or Java objects. Utilizing the PSP mechanism, user interfaces can be created which run separately from the cluster society. Essentially, the user is writing a web application when using the PSP mechanism.

Cluster Relationships

Clusters form *relationships* with each other, establishing *capabilities* and *roles* in the process. At a minimum, each cluster is required to have an "Administrative Superior" cluster. The exception to this is the one cluster that resides at the top of the cluster hierarchy. During the cluster's startup phase, ALP establishes an Administrative Superior/Subordinate relationship between the cluster being initialized and its superior. Then, while the clusters are processing, each one will have a reference to the other in its list of Organization assets. Moreover, there will be a role associated with the reference, in this case, either "AdministrativeSuperior" or "AdministrativeSubordinate." In addition to this automatically generated relationship, clusters

can selectively establish relationships and roles with each other. These relationships can also include capabilities. For example, cluster A may have cluster B as a “supporting” cluster with “division supply provider” capability. In this example, B has a role of “supporting” cluster A, and A sees B as having the capability of “division supply provider.” When an allocator in cluster A assigns resources to subtasks, it might decide to redirect subtasks to cluster B when a “division supply provider” allocation would be appropriate.

Putting the Pieces Together

Figure 2 gives a simple linear representation of plugin activity as a cluster and its plugins process a single task. After a cluster receives a task, it is passed to an expander plugin. The expander creates a workflow of subtasks and submits it to the LogPlan. The allocator plugin gets the workflow of subtasks, finds assets to allocate to them (or perhaps assigns tasks to another cluster), calculates values for the allocation result, and submits the results to the LogPlan. The ALP infrastructure creates notification information to pass back to the cluster that made the original request. An assessor plugin may also review the results of the allocations and generate additional directives.

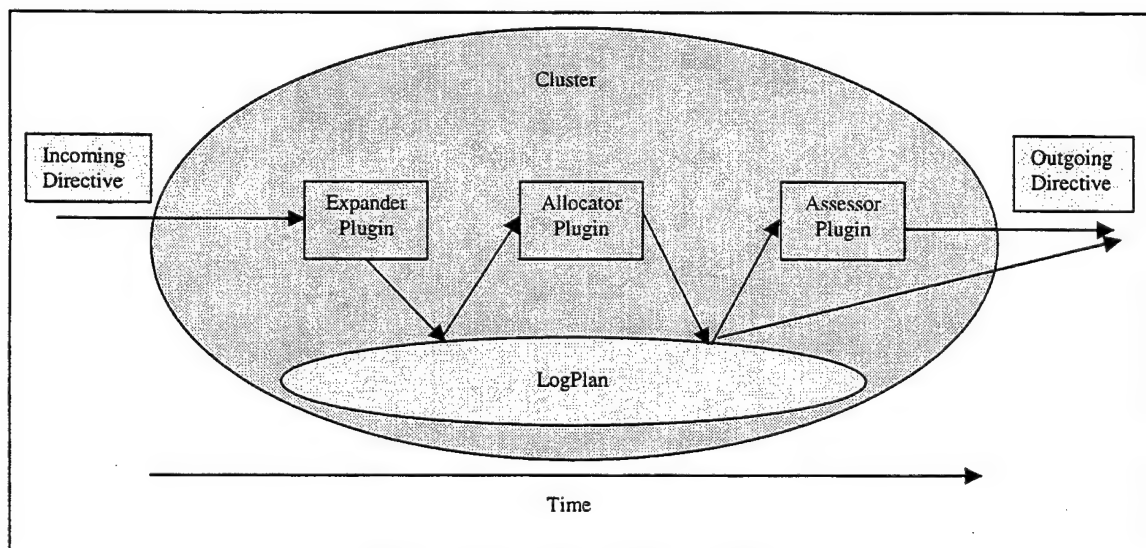


Figure 2: Plugin Operation Flow

ALP's Decision-Making Philosophy

ALP is based on a decision-making philosophy focused on providing solutions that continually improve tolerances rather than attempting to initially provide a "best" solution. The motivation for this is the highly distributed nature of the ALP architecture. To achieve a "best" solution would require a centralized location to request everything of every provider, then decide for everyone, who gets what and when. But, in a highly distributed system, a centralized location containing all the rules and logic does not exist. Even if it did, the amount of data required would be prohibitive. ALP provides a different solution. Each task is created with an associated *penalty function*; a penalty function can be thought of as a set of thresholds. Recall from the previous section that an allocator supplies values in an allocation result when it allocates a resource. The requester of the task will get that allocation's result values and will pass it to the task's penalty function. If the penalty function indicates the penalty value is "acceptable," the cluster could just accept the situation and continue. If it is "unacceptable," the cluster could rescind the task and request a different cluster. This assumes there are other clusters available to do the task; otherwise, "unacceptable" may necessarily be accepted. If the penalty function indicates a "borderline" condition, the cluster could keep the allocation but start creating additional tasks to do some comparative shopping. If the cluster finds a more acceptable allocation from a different source, it could keep the alternate allocation and rescind the original request.

Note that this methodology focuses on keeping all of the elements of a plan within tolerance levels. This approach vastly reduces the number of requests that have to be passed from cluster to cluster, resulting in fewer burdens placed on the communications processes. Also, note that this is where assessor plugins can play an important role. These plugins could generate low-priority requests that are intended to find alternative solutions to improve tolerance levels, but could be processed during "lower" activity times.

ALP Development Environment

The ALP architecture development team elected to use Java and Java-based tools for the development of the ALP architecture. Java provides the platform independence and includes

powerful networking and security capabilities as a part of the language. The current release of the ALP architecture utilizes Sun's Java Development Kit (JDK) version 1.2.2.

The development team for this effort also utilized Inprise's Java integrated development environment (IDE) named JBuilder 4 for constructing the demonstration software. Pentium III - based personal computers running Microsoft's Windows NT or Windows 2000 operating system were the development machines used. Although a personal computer/Windows configuration was utilized, cross platform capability was accomplished as a result of doing all development utilizing the Java programming language. The demonstration software was successfully executed on a Linux platform as well.

Yearly Demonstration Scenarios

One of the objectives of the ALP program is to provide a yearly flag-level demonstration. These demonstrations, each based on a fictitious contingency scenario, showcase the achievements of each year's technical development.

The first year's demonstration consisted of about a dozen clusters, each representing a different military organization. These clusters were distributed across a wide-area network. The objective of this demonstration was to prove the feasibility of the ALP infrastructure to communicate tasks between remotely located clusters, to propagate the results of these tasks and to demonstrate that logistics planning goals and requirements could be achieved effectively through ALP's messaging syntax and protocols.

The second year's demonstration consisted of approximately 50 clusters. These clusters represented organizations from the Army and the Air Force as well as from DLA and TRANSCOM. The objective of this demonstration was to show ALP's ability to plan for a more accurately sized deployment scenario in terms of number of persons and assets represented and allocated. Also, the demonstration began to focus on fidelity of processes in supply chains and transportation activities.

The third year's demonstration consisted of several hundred clusters. From this demonstration, ALP's ability to perform as a robust and scalable planning tool began to surface. This demonstration showcased ALP's ability to plan for multiple contingencies simultaneously, to effectively handle perturbations to the plan and to respond to real-world events during plan execution, automatically adjusting the plan in order to maintain a working plan consistent with original operational objectives.

The last demonstration is expected to consist of over a thousand clusters. These clusters will accurately represent the planning processes of the organizations they model. In particular, the entire supply chain for numerous classes of supply, including fuel, subsistence, medical and spare parts, will be represented with very high fidelity and will be a focal point for the demonstration.

Future of ALP

The ALP program is entering its final year of research and development. It is generally believed that the ALP distributed agent technology has proven to be a thorough logistics planning system capable of continual, automatic and effective replanning due to plan perturbations and execution monitoring. This final year of development is focused on filling in a number of details, repackaging the open-source version of the technology, called COUGAAR, and on creating general-purpose tools and plugins for cluster/society development.

Another DARPA program is picking up where ALP has left off. This program, called Ultra*Log, is a four-year program that will be based on ALP's distributed agent technology. ALP demonstrated that distributed agent architecture technologies could be applied to the logistics domain to maintain total control of the logistics pipeline. However, the power of this agent technology to fuse vast amounts of data makes it vulnerable in the information warfare environment of the future. Also, the capability of this technology to monitor and react to real world changes that occur frequently during wartime makes it susceptible to chaotic behavior. The nature of the Cougaar technology to model and automate business processes makes it critical to logistics support of war fighting operations. For all of these reasons, a fully instantiated ALP-based logistics society lacks the survivability (security, scalability and robustness) to deal

effectively with wartime environments involving heavy information warfare and kinetic attrition components.

The Ultra*Log project will pursue the development of technologies to enhance the security, robustness, and scalability of large-scale, distributed, agent-based systems operating in chaotic wartime environments. The objective of the Ultra*Log project is to pursue leading-edge technologies in these three areas to create comprehensive capability which will enable a massive scale, trusted, distributed agent infrastructure for operational logistics to be survivable under the most extreme circumstances.

AEF COMMUNITY – LOGISTICS PLANNING FOR AEF DEPLOYMENTS

Development Progression of the AEF Community of Clusters

The Air Expeditionary Force (AEF) Community of clusters has been under development for approximately three years. During the first year of the ALP program, AFRL/HESR representatives attended an ALP briefing. After attending this briefing, AFRL/HESR personnel decided to get involved in the ALP project and drafted a Statement of Work (SOW) that was focused at creating a wing-level cluster society based on the ALP infrastructure. Since this initial effort, two additional follow-on efforts have been executed. AFRL/HESR was interested in the applicability of the ALP infrastructure to support the demanding logistic requirements imposed by the AEF concept of operations. The following paragraphs will describe the progression of the AEF cluster community over the three years of development.

During the first year of development, a small cluster society consisting of approximately 10 clusters was developed. This first AEF cluster community modeled a single AEF consisting of three fighter wings and a single hypothetical provisional wing. Additionally, a TRANSCOM cluster and a War Reserve Material (WRM) cluster were created. One emphasis of the initial AEF cluster community was to allow the user to participate in the decision making process modeled in ALP. The squadron user interface allowed the user to override the selection of aircraft for the deployment. Figure 3 provides a cluster diagram of the first AEF cluster

community. Clusters are represented by ovals, lines drawn between the clusters identify inter-cluster relationships. For a more complete description of the initial AEF cluster community, please refer to the technical report titled "Development of an Air Force Wing Level Logistics Cluster for use with the Advanced Logistics Project Architecture" cited in the reference section at the end of this report.

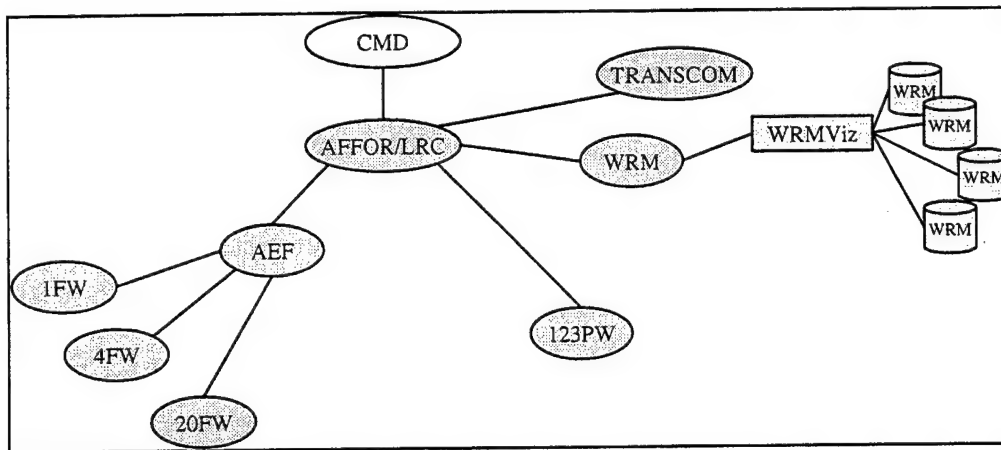


Figure 3: Year One AEF Cluster Community

The second year of development resulted in a much larger and sophisticated AEF cluster community. This phase of development consisted of the largest funding profile over the three years of development. The community grew to over 40 clusters and much more detailed decision making logic was added. The following bullets highlight the major accomplishments of the second year of development on the AEF cluster community.

- More detailed representation of the organizations involved in the AEF deployment process.
- More detailed modeling of the AEF vision of AEF deployment processes.
- Changed from using flat files to relational databases in order to efficiently store and retrieve the needed data to support the AEF cluster society.
- Added support for dynamic replanning as a result of deviations from the original plan. Three different dynamic replanning scenarios were supported in the AEF society.
- Interaction with a non-ALP external system called WRMViz. This interaction allowed the AEF community to query on the status of equipment maintained at War Reserve Material locations.

- Interaction with the DLA cluster community for handling requests for fuel and spare parts sustainment tasks.
- Enhanced interaction with the TRANSCOM cluster community including a response mechanism from the TRANSCOM cluster community back to the AEF cluster community.

Figure 4 provides a cluster diagram of the second year AEF cluster community. Three of the 10 AEF organizations and five different OCONUS wings were modeled. For a more complete description of the second year development of the AEF cluster community, please refer to the technical report titled “Air Force ALP AEF Initiative Wing-Level Cluster Development and Demonstration” cited in the reference section at the end of this report.

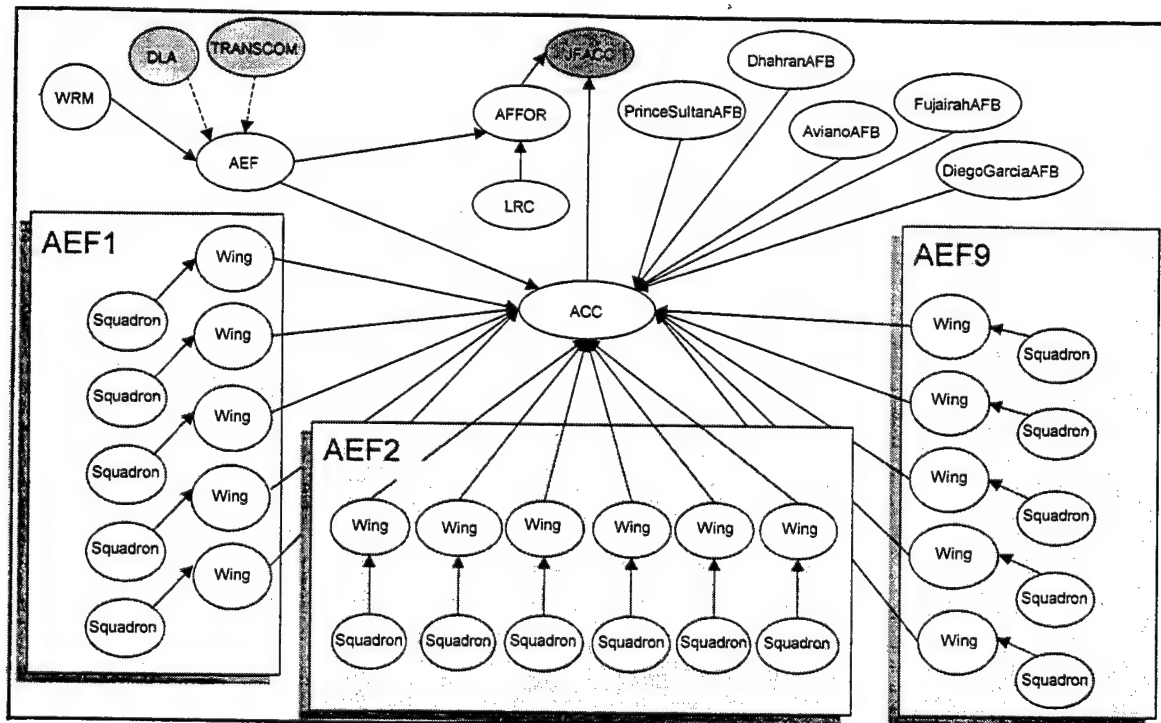


Figure 4: Year Two AEF Cluster Community

The third year of development resulted in an AEF cluster community consisting of over 100 clusters. There were three primary focus areas to be addressed during this year of the development. These areas included: support for multiple OPLANS, “what-if” planning and the

dynamic build up of logistic requirements based on number and type of aircraft, mission objectives, and beddown location. Details on how these areas were addressed will be covered in the section titled "Synopsis of this year's work". One main difference in this society versus the previous year is the addition of the sub squadron clusters. This was done to model the portion of the squadron that was deploying to support a particular AEF. It also allowed for the projection of required fuel and spare parts requirements for the portion of the squadron deploying. Another noticeable difference is that the beddown locations each consist of three clusters. The addition of clusters to support spare parts inventory management and fuel inventory management were added to each of the beddown configurations. Finally, after reviewing AEF doctrine and visiting the AEF Center at Langley AFB, additional clusters were added including the XO, XOP, XOPW, and the AEFC clusters. More details on the organizations modeled will be covered in a later section titled "Descriptions of Organizations Modeled in the AEF Community." Figure 5 provides cluster diagram of the current AEF cluster community.

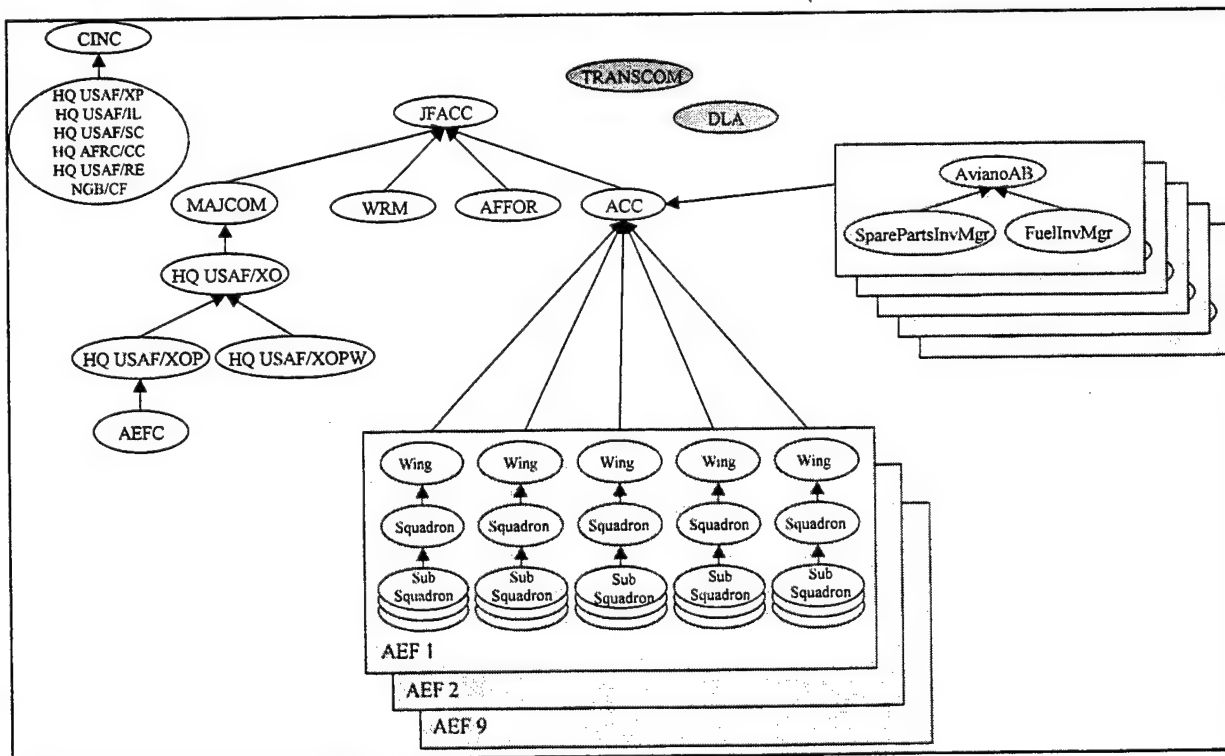


Figure 5: Year Three AEF Cluster Community

Synopsis of this Year's Work

This section will discuss the goals and accomplishments for the work completed on this delivery order. As with prior year efforts, a SOW was drafted that was tightly coupled with the anticipated progression of the underlying ALP infrastructure upon which the AEF cluster community is built. Being dependent on the progression of the infrastructure resulted in some deviations from the SOW. One such deviation was that support for handling different "what-if" scenarios was to be added. Developing this functionality was very dependent on infrastructure interfaces that were not completed by ALPINE. The ALPINE development team determined the "what-if" interface was much more involved than originally anticipated and it was decided to delay its implementation until the following year. The team also took on additional tasking at the request of DARPA program management. Most notably this included supporting the Air Force Institute of Technology's (AFIT) ALP initiative.

Descriptions of Organizations Modeled in 2000 AEF Community

The AEF community developed under this year's efforts consists of over 100 clusters. Much of the functionality from previous efforts was used as a starting point for this effort. This includes aircraft and personnel selection capabilities on Wing and Squadron clusters and equipment sourcing functionality used by all clusters. A major effort under this contract was devoted to a more thorough modeling of several key command organizations involved in AEF deployment processes. Clusters were developed to capture essential functionalities representing the HQ USAF/XOP, HQ USAF/XOPW and the AEFC organizations. In particular, HQ USAF/XOP plays a central role in selecting a successful force package mix and in the determination of specific squadrons for missions as well as a launching point for equipment sourcing and personnel sourcing tasks. The AEFC plays a central role in the sourcing loop. One of its key responsibilities is to mediate conflicts arising from shortfalls, working with the MAJCOM to resolve issues and establish priorities. The HQ USAF/XOPW is a key organization for the establishment of standardized equipment unit type codes (UTCs), so it works with the associated component in the Mission-Resource Value Assessment Tool (M-R VAT) tool for incorporating time-phased equipment requirements into the AEF planning scenario. Figure 6 shows how these organizations interact.

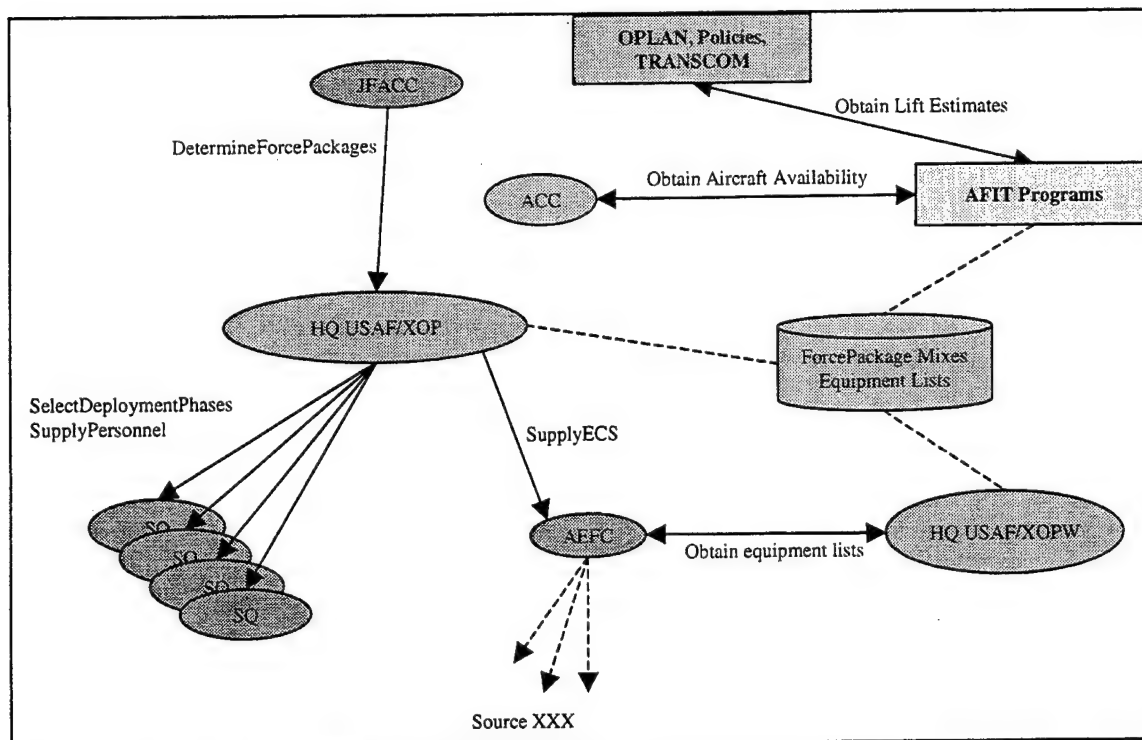


Figure 6: Key Components in the AEF Community

High Level Task Flow in the 2000 AEF Community

The elemental basis of the ALP development effort is to make critical, essential improvements in logistics planning and execution. Clusters within ALP can interact with systems external to ALP. This interaction can provide knowledge to ALP and to the external system in such a way that both systems benefit from the relationship, and the resulting knowledge and decisions are expertly adapted in a cohesive, synergistic method.

The ALP AEF Community has developed a very good example of this type of relationship in an interface between the ALP AEF Community and the M-R VAT, a system external to ALP, developed by the Air Force Institute of Technology (AFIT). The purpose of M-R VAT is to provide a methodology for rationally assigning relative value to material resources over time in order to improve the linkage between what arrives in theater on any given day and what is needed in theatre on any given day. The ALP AEF Community utilizes the knowledge held in M-R VAT, combined with the ALP AEF Community knowledge of realistic logistical feasibility

to provide an optimum solution within real-world constraints for time-phased availability of contingency resources.

As previously stated, in the ALP Community, all 'work' is accomplished through creating tasks and performing work to satisfy these tasks. In the ALP AEF Community, the tasking names are defined in a descriptive manner to accurately identify and model the real-world processes and actions. The cluster functionality is also modeled in such a way to represent an accurate view of the Air Force protocols and lines of authority.

Refer to Figure 6 for the following workflow discussions.

The process workflow for the M-R VAT tool:

- The M-R VAT tool requests available aircraft resource information from the ALP AEF.
- Based on this available aircraft resource information, lift information from the TRANSCOM Community, high-level objectives and directives from the CINC Community, and Planning Operator preferences, M-R VAT computes a set of prioritized force package mix data, which includes data such as aircraft types, missions and sortie rates as well as time-phased requirements for equipment.
- This data is then made available provided to the ALP AEF Cluster Community.

The high-level task flow for the ALP AEF Cluster Community:

- The ALP AEF Cluster Community is initiated and the JFACC Cluster is tasked to "GetLogSupport" for a specified Oplan.
- The JFACC Cluster then creates a subtask called 'Determine Force Packages' and sends this tasking to the HQ USAF/XOP Cluster.
- The HQ USAF/XOP Cluster extracts the force package mix data supplied from M-R VAT, and commences to try to satisfy a force package mix, beginning with the highest priority force package mix. Using the AEF rotation knowledge, which provides the time-phased availability of the AEF aircraft resources, the HQ USAF/XOP Cluster now attempts to locate squadrons with the appropriate capabilities to satisfy this force package

mix. This is accomplished by sending a 'Select Deployment Phases' task to appropriate squadron clusters.

- When all the requested aircraft have been identified, and have reported that they are indeed available during the prescribed period, the specific force package mix is deemed a 'success', and a sequence of tasks are now created to supply the additional resources needed to satisfy the high-level task of 'Determine Force Packages'. The high-level tasks needed to accomplish this are:
 - A task sent to the AEFC Cluster to 'Supply Expeditionary Combat Support'.
 - A task sent to the AEFC Cluster to 'Supply Beddown Support'.
 - Tasks sent to the appropriate squadrons clusters to 'Supply Personnel'.
 - Tasks sent to squadrons to generate sustainment requirements.

The AEFC Cluster tasks then break down into:

- A task sent to the HQ USAF/XOPW Cluster to 'Obtain Appropriate Equipment'.
Functionality in the processing of this task interacts with more data provided by the M-R VAT tool. In particular, time-phased equipment lists associated with the "successful" force package mix are obtained from M-R VAT output. These equipment requirements are then sourced through ALP mechanisms.
- Tasks sent to the Beddown Base Clusters to 'Supply Beddown Support'.

Once a force package mix is successful, the ALP AEF Cluster Community reports the success of the tasks, and the user can view the results through the ALP AEF Force Package Mix User Interface. From this interface, the user can also make changes to the force package mixes and feed these changes back to the AEF community for replanning. This capability is discussed further in the sections "User Interfaces" and "Operations and Logistics Synergy".

Operations and Logistics Synergy

Over the course of the ALP program it has become more and more important to develop capabilities that bring the Operations Community and the Logistics Community closer together. One of the goals of AEF community development under this effort was to demonstrate this synergy by using mission details from the Operations community as inputs for the ALP logistics

planning processes. ALP then generates a logistics plan and feeds results back to the Operations community. Using this feedback, Operations can then make adjustments to their mission planning and submit these changes back to the ALP community for replanning. Figure 7 shows how this information is intended to flow.

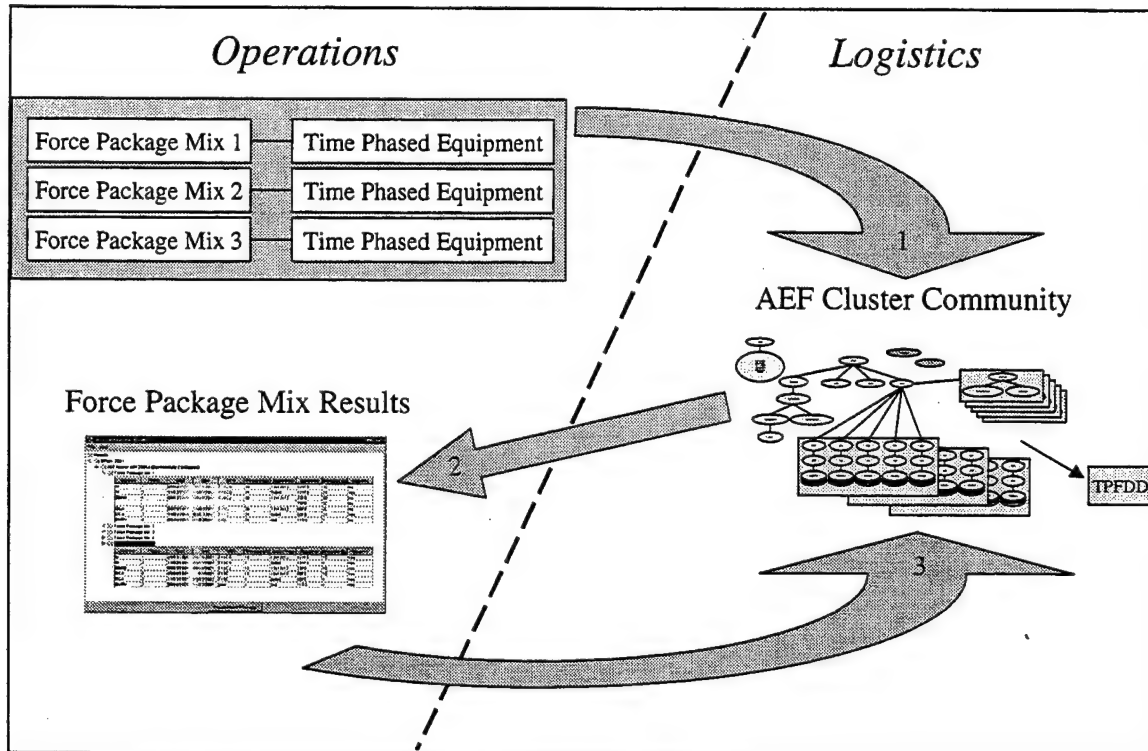


Figure 7: Ops/Log Synergy

User Interfaces

Several user interfaces were developed under this effort to expose the planning results of the AEF community and to engage the user for feedback in these planning processes.

As discussed in the section "Operations and Logistics Synergy", one of the functional features of this year's AEF community is the ability to provide logistics planning results to the Operations community and to solicit feedback resulting in replanning. Figure 8 shows a user interface that is available after the AEF community initially plans for a scenario.

HQ-USAF-XOP Force Package Mix Results

File

Results

Opplan: 2001

AEF Name: AEF2001A

Force Package Mix: 1 (Fail)

Mission	MDS	Desired Qty	Start	End	Sortie Rate	Squadron	Allocated Qty	Success
AI	F16CG	36	08/14/2005	09/03/2005	12x12x12	421FS	30	No
AI	F16CG	24	10/14/2005	02/10/2006	8x8x8	421FS	24	Yes
SEAD	F16CJ	36	08/14/2005	09/03/2005	12x12x12	23FS	30	No
						79FS	30	No
OCA	F15C	36	09/04/2005	10/13/2005	12x12x12	60FS	30	No
OCA	F15C	24	10/14/2005	02/10/2006	8x8x8	60FS	24	Yes
SURV	UAV	24	09/04/2005	10/13/2005	8x8	11RS	16	No

Force Package Mix: 2 (Fail)

Force Package Mix: 3 (Success)

Mission	MDS	Desired Qty	Start	End	Sortie Rate	Squadron	Allocated Qty	Success
AI	F16CG	18	08/14/2005	09/03/2005	12x12x12	421FS	18	Yes
AI	F16CG	12	10/14/2005	02/10/2006	8x8x8	421FS	12	Yes
SEAD	F16CJ	18	08/14/2005	09/03/2005	12x12x12	23FS	18	Yes
OCA	F15C	18	09/04/2005	10/13/2005	12x12x12	60FS	18	Yes
OCA	F15C	12	10/14/2005	02/10/2006	8x8x8	60FS	12	Yes
SURV	UAV	12	09/04/2005	10/13/2005	8x8	11RS	12	Yes

Figure 8: Force Package Mix Results

From various force package options originally supplied by planning efforts within the operations community – in this case, from AFIT's M-R VAT tool the user is able to see individual missions and components that succeeded and those that failed. By right-clicking on one of the displayed tables, the user can use this feedback to generate details for an alternate force package mix, then submit those details back to ALP to replan the campaign. See Figure 9.

Force Package Mix Editor

Mission	MDS	QTY	Start	End	Sortie Rate
AI	F16CG	18	08/14/05	09/03/05	12x12x12
AI	F16CG	12	10/14/05	02/10/06	8x8x8
SEAD	F16CJ	18	08/14/05	09/03/05	12x12x12
OCA	F15C	18	09/04/05	10/13/05	12x12x12
OCA	F15C	12		02/10/06	8x8x8
SURV	UAV	12		10/13/05	8x8

Insert Row Before
Insert Row After
Copy Row
Paste Row
Delete Row
Move Row Up
Move Row Down

Commit New Force Package Mix Cancel

Figure 9: Create Force Package Mix

Figure 10 shows a user interface that would be available at each squadron. This view shows a Gantt chart of allocation schedules for each aircraft. The right hand side of this view identifies all of the aircraft available at the particular squadron.

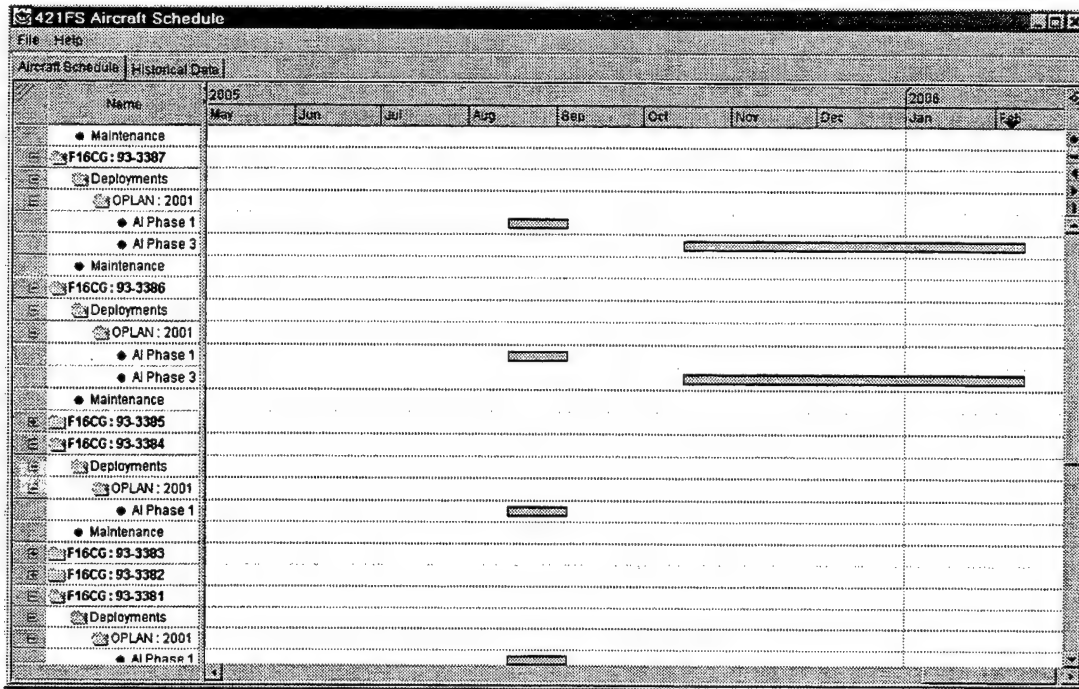


Figure 10: Squadron Aircraft Schedule

Another view available on this application display details on flight history, projected maintenance schedules and operational availability (Figure 11). This view is accessed by selecting the "Historical Data" tab on the squadron window.

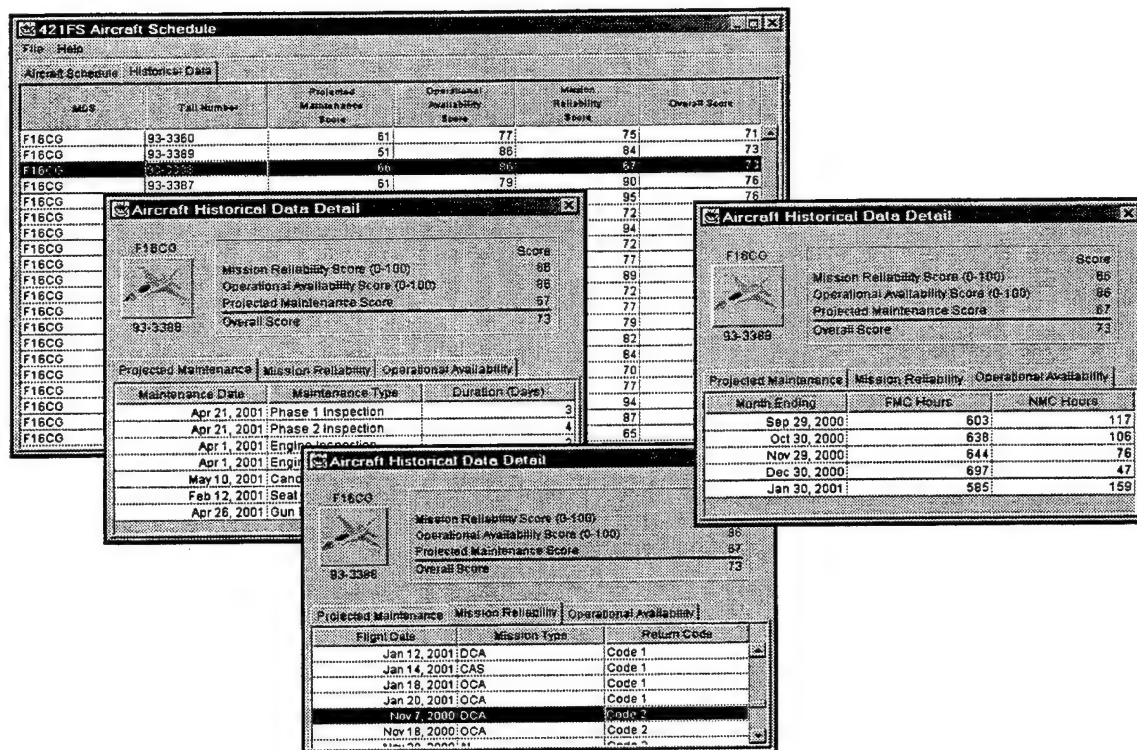


Figure 11: Aircraft Historical Data

Figure 12 shows a user interface that provides a detailed listing of all transportation information including scheduled transport tail numbers, detailed itinerary and manifest. This application provides a detailed textual view and a Gantt view of the transportation planning results.

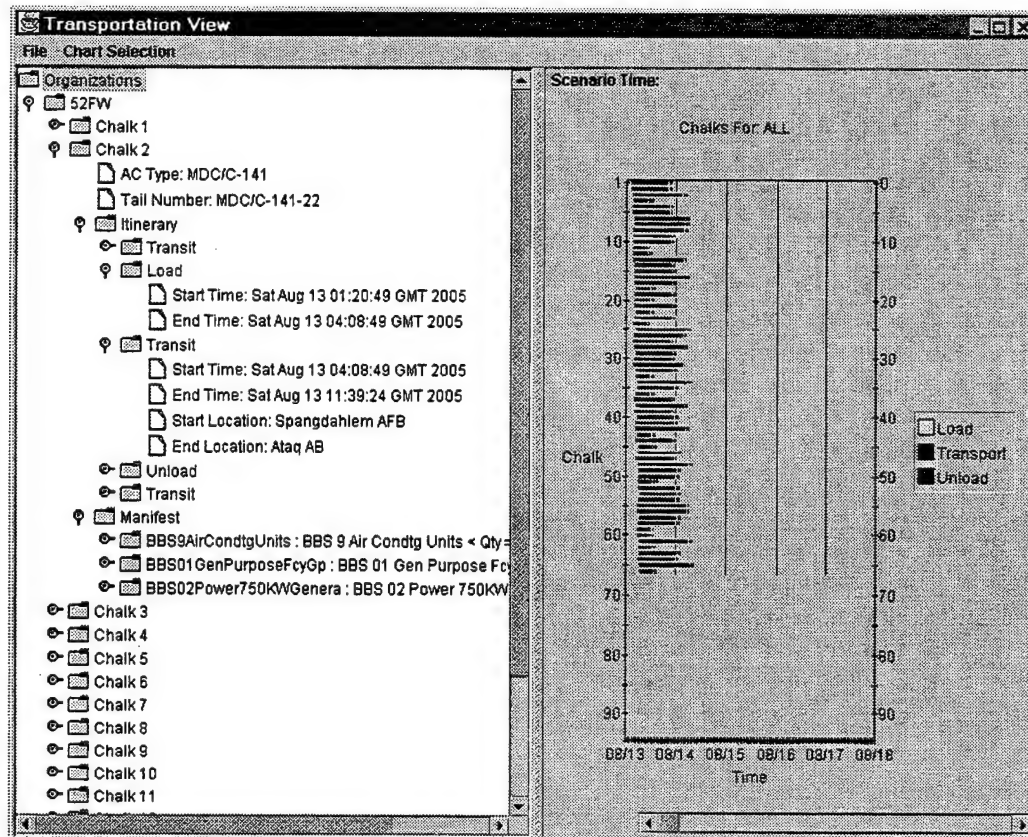


Figure 12: Chalk View User Interface

Figure 13 shows a browser-based user interface created to aid efforts to integrate the AEF community into the entire demonstration society. During society integration it is useful to have interfaces into the various communities that provide information about the planning processes going on in those communities. The AEF Society Completion User Interface provides details about the planning processes in the AEF community. The user can choose to collect and display this information from the HQ USAF/XOP cluster or from the AEFC cluster. The information can be presented in complete detail or can be presented in summary format.

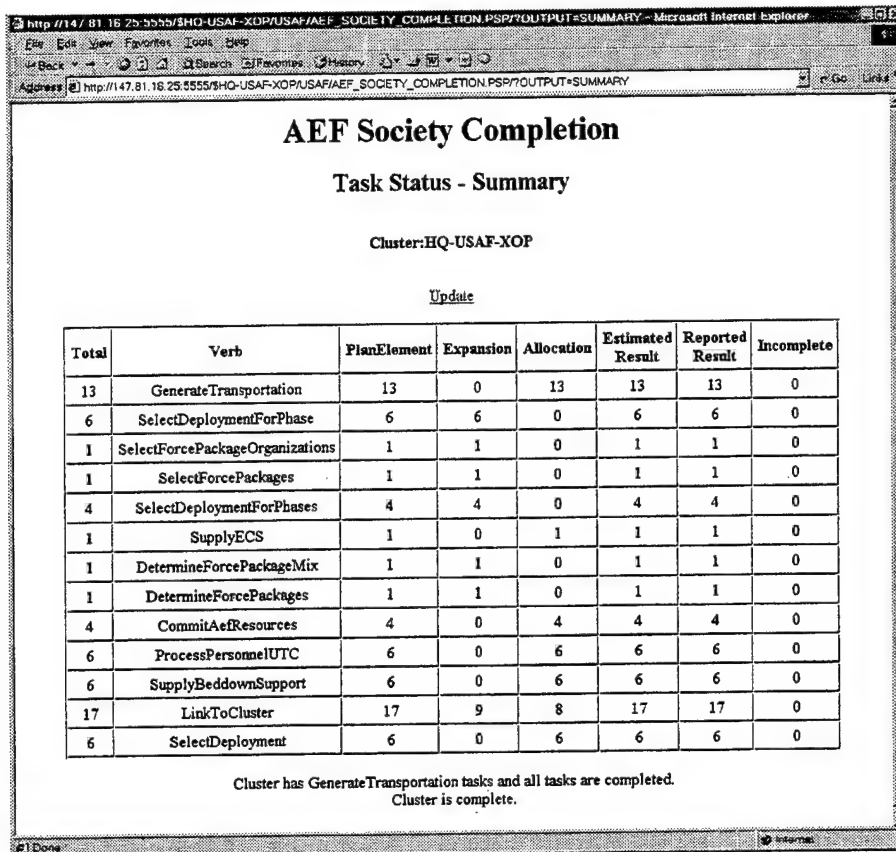


Figure 13: AEF Society Completion User Interface

Figure 14 shows another browser-based user interface. This interface displays all the equipment requirements for each deploying AEF. The total number requested of each NSN, successfully allocated and any resulting shortfalls are displayed on this user interface.

http://localhost:5555/AEFCAUSA/AEF_SOCIETY_ASSET_SOURCING_RESULTS.PSP - Microsoft Internet Explorer

Address: http://localhost:5555/AEFCAUSA/AEF_SOCIETY_ASSET_SOURCING_RESULTS.PSP

AEF Society Asset Sourcing Results Report

[Update](#)

Opplan: 2001, AEF Name: AEF2001A

NSN	Nomenclature	Requested	Allocated	Shortfalls
6230012240938	Floodlight Set	67	63	4
TASC3333333344	SAR Equipment	1	1	0
1560010410546	Hydrazine Casket	4	4	0
4930012488719	Howard HYD Dispenser	4	3	1
3920001654135WR	F-2 Trailer Loaded	13	13	0
1730005037625	Support H/B	1	1	0
4310013706351	MC-2A W/Shoring	4	3	1
3655010385125	LN2 Cart	4	4	0
1740007135908	3000 TR	42	39	3
1730014896456	MHU 83 Fork Adpt Assembly	1	1	0
1730010904984	Dolly Fuel Tank	4	4	0
1730011805744	CFT Maint Stand	1	1	0
1730005162019	463L PLT	1	1	0
BBS01Kitchen-Dining9-1	BBS 01 Kitchen-Dining 9-1	90	83	7
3950010933721RC	Regency Crane	1	1	0
1730002948884	B-5 Stand	4	3	1
1730002948883	B-4 Maint Stand	23	22	1
3655005402733	GOX Cart	4	3	1

Figure 14: Asset Sourcing Results User Interface

GENERIC FUNCTIONALITY AND INTEGRATION WITH OTHER ALP COMMUNITIES

This year's AEF community now takes advantage of generic features and plugins made available by other subcontractors. In particular, the squadron clusters now incorporate plugins that utilize the ICIS model to generate sustainment projections for fuel and spare parts. Also, the AEF community now contains clusters at beddown locations for fuel and spare parts inventory management. The plugins in these clusters are tailored versions of generic plugins available to all society builders.

To further incorporate the AEF community into the entire demonstration society, this year's AEF community is fully integrated with the TRANSCOM community and the DLA community.

Generalized Tools and the Generic Logistics Model (GLM)

It became apparent early in the ALP program that general, standard building blocks and tools for cluster and society development as well as generic, sharable plugins need to exist for ALP to be successful. Primarily, this is necessitated due to the potential size and complexity of an ALP society along with its highly distributed nature. An ALP society could conceivably contain thousands or millions of clusters, distributed worldwide. It is not realistic or economical to expect that the content of these clusters would each be custom developed. The development costs would be prohibitive and system maintenance would be impossible. Fortunately, the various components of an ALP-based system have a number of common features and functionality. In particular, the clusters in an ALP-based logistics system will typically need to maintain complex schedules, deal with multiple suppliers of resources, manage inventories, etc. It makes sense that generic, tailorable tools and plugins should be developed to consistently and reliably handle these processes. These logistics related generic tools and plugins are referred to as the Generic Logistics Model (GLM).

During the last several years of the ALP program there were numerous initiatives to create more generalized tools and plugins for the GLM. These efforts ranged from the creation of scriptable functionality for plugins to the development of flexible, tailorable plugins for specific logistics related problems.

Description of Generic AEF Functionality

The original development of the AEF community of clusters was written to meet the specific needs of the Air Force organizations being modeled. As the ALP program progressed, much of this functionality was modularized and generalized for availability to other cluster developers. This generic functionality included mechanisms for inter-cluster relationship management and for prioritized asset selection functionality.

Contributions to the GLM

During this effort, several pieces of AEF functionality were generalized and submitted for inclusion in the GLM. The most significant of these were a set of plugins that effectively manage

the use of multiple suppliers. Our GLM submission included documentation, tailoring instructions and a demonstration society. These plugins are called SourceExpander and SourceAllocator; the plugins can be used to provide support for situations in which a cluster has more than a single supplier that could be used for a Supply task. The following example describes the kind of situation appropriate for these plugins.

Example: Suppose a cluster is tasked to supply 10 generators. Further suppose the cluster has several "provider" Organization assets, possibly including the cluster itself, which could be used to supply these generators. It would be nice to have generic functionality that would ask one of these providers to supply the 10 generators. If that provider's AllocationResult indicates it can supply 6 of the desired 10, the generic functionality would then ask one of the other providers for the remaining 4 generators. This process would continue until all of the desired generators have been acquired, or until there are no more providers.

In addition to the functionality described in the example, it would be beneficial if the list of providers could indicate a preferred order of use for the suppliers. Moreover, the list of suppliers may be dependent on the item being supplied, or on characteristics of the task, mission or OPLAN associated with the item's use.

Also, the plugins supporting these multiple suppliers should be able to handle dynamic changes in the AllocationResults of the various suppliers, adjusting the requests to alternate suppliers as needed.

All of these features are supported by the Multiple Suppliers plugins. See the Multiple Suppliers PlugIn Documentation for further details of these plugins and their uses.

In addition to the Multiple Suppliers PlugIns, we also submitted other tools for inclusion in the GLM. These include numerous mechanisms for interacting with the LogPlan, date manipulations, and a powerful user interface for developers that displays LogPlan contents.

FINAL YEAR OF ALP DEVELOPMENT

The development of the AEF community of clusters has been ongoing for the past three years. During the final year of the ALP program, very little new development will be done on the AEF

community of clusters. It has been determined that the AEF community has evolved to a point where the feasibility of utilizing the ALP infrastructure for modeling various aspects of the Air Forces' logistic domain has been realized. An effort is underway to try and identify an Air Force pilot program focused on applying the ALP technology. This "to be determined pilot program" could utilize some or all of the AEF community development accomplished over the years as a starting point.

To support the final ALP demonstration in May 2001, a minimal amount of work will be conducted on the AEF cluster community. This will include maintaining compatibility with the future releases of the infrastructure. Additionally, support will be provided to ensure the successful integration of AFIT's M-R VAT tool with the AEF cluster community.

CONCLUSIONS

The ALP program has performed and delivered capabilities meeting and exceeding expectations over the course of the last several years. ALP's underlying architecture and associated technologies does provide the capability to fundamentally improve logistics planning and execution while effectively providing logisticians, war planners and war fighters the ability to plan, execute, monitor and replan in real time.

The ALP system is distributed and highly automated. Being distributed, domain knowledge and functionality can actually be maintained and run local to the domain it represents. On the other hand, distributed systems have the potential problem in which the distributed agents are oblivious of each other, unable to effectively leverage each other's capabilities. But ALP agents have the ability to publish their roles and capabilities, so appropriate inter-agent relationships can be determined and established while the system is running. Being automated allows domain knowledge, business rules and expertise to be captured and incorporated in the ALP system, vastly decreasing the time it takes to develop, execute and maintain operational and logistics plans.

The ALP system has effectively created a standard communications protocol while supporting the ability to interact with other non-ALP systems.

Perhaps most importantly the ALP system supports continuous planning/replanning, providing timely logistics details, empowering logisticians, operations and war planners to effectively, accurately and quickly accomplish plan objectives. This detail provides a richness of fidelity and flexibility by taking advantage of ALP's Penalty Functions and Allocation Response mechanisms.

The development of the AEF community of clusters has demonstrated that systems built upon the ALP agent technology can effectively solve logistics problems in a scalable and timely manner. These systems can respond to changes in real world situations, automatically replanning in order to maintain a stable and consistent plan that adheres to original plan objectives. The AEF community has shown that, although ALP is an "automated" system, it effectively provides user interfaces that incorporate ALP's high level of planning detail in decision support applications. Users can then interact with ALP's planning processes, making adjustments and validating results as well as policies and decision-making rules.

The infrastructure of ALP has several significant goals yet to accomplish, and these are being addressed under the Ultra*Log follow-on program. In particular, scalability on a worldwide scale, robustness in wartime scenarios and security at a DOD level are issues being taken up in Ultra*Log. In addition, the final year of the ALP program and the Ultra*Log program will both be addressing the ability to generate "what-if" scenarios.

Overall, the ALP program has proven to be a great success, developing capabilities beyond expectations. DLA is now fielding applications built on ALP technology and several pilot programs within the DOD are under way. ALP's infrastructure, tools and training materials have gone Open Source along with an associated web site www.cougaar.com, which has already sparking additional interest from the private sector.

REFERENCES

1. Allen, Christopher S., Patrick K. Clark, Nicholas J. Stute, and Christopher K. Curtis. "Air Force ALP AEF Initiative Wing-Level Cluster Development and Demonstration." AFRL-HE-WP-TR-2001-0037, January, 2001.
2. Allen, Christopher S., Patrick K. Clark, Nicholas J. Stute, and Christopher K. Curtis. "Development of an Air Force Wing Level Logistics Cluster for use with the Advanced Logistics Project Architecture." AFRL-HE-WP-TR-1999-0225, March 1999.
3. ALPINE Joint Venture Document. "Advanced Logistics Project (ALP) PlugIn Developers' Guide" Version ALP-PDG 6.6.
4. Advanced Logistics Project (ALP) Cougaar Architecture Document (ArchDoc), Version 6.4, ALPINE, August 16, 2000.
5. Ultra*Log BAA PROPOSER INFORMATION PAMPHLET (PIP), DARPA.